An Efficient Reduction of Ranking to Classification

Nir Ailon
Google Research
76 Ninth Ave, 4th Floor
New York, NY 10011
nailon@google.com

Mehryar Mohri
Courant Institute and Google Research
251 Mercer Street
New York, NY 10012
mohri@cims.nyu.edu

Abstract

This paper describes an efficient reduction of the learning problem of ranking to binary classification. The reduction is randomized and guarantees a pairwise misranking regret bounded by that of the binary classifier, improving on a recent result of Balcan et al. (2007) which ensures only twice that upper-bound. Moreover, our reduction applies to a broader class of ranking loss functions, admits a simple proof, and the expected time complexity of our algorithm in terms of number of calls to a classifier or preference function is also improved from $\Omega(n^2)$ to $O(n \log n)$. In addition, when the top k ranked elements only are required ($k \ll n$), as in many applications in information extraction or search engine design, the time complexity of our algorithm can be further reduced to $O(k \log k + n)$. Our reduction and algorithm are thus practical for realistic applications where the number of points to rank exceeds several thousands. Much of our results also extend beyond the bipartite case previously studied. To further complement them, we also derive lower bounds for any deterministic reduction of ranking to binary classification, proving that randomization is necessary to achieve our reduction guarantees.

1 Introduction

The learning problem of ranking arises in many modern applications, including the design of search engines, information extraction, and movie recommendation systems. In these applications, the ordering of the documents or movies returned is a critical aspect of the system.

The problem has been formulated within two distinct settings. In the *score-based setting*, the learning algorithm receives a labeled sample of pairwise preferences and returns a *scoring function* $f : U \to \mathbb{R}$ which induces a linear ordering of the points in the set U. Test points are simply ranked according to the values of f for those points. Several ranking algorithms, including RankBoost (Freund et al., 2003; Rudin et al., 2005), SVM-type ranking (Joachims, 2002), and other algorithms such as PRank (Crammer & Singer, 2001; Agarwal & Niyogi, 2005), were designed for this setting. Generalization bounds have been given in this setting for the pairwise misranking error (Freund et al., 2003; Agarwal et al., 2005), including margin-based bounds (Rudin et al., 2005). Stability-based generalization bounds have also been given in this setting for wide classes of ranking algorithms both in the case of bipartite ranking (Agarwal & Niyogi, 2005) and the general case (Cortes et al. 2007b; 2007a).

A somewhat different two-stage scenario was considered in other publications starting with (Cohen et al., 1999), and later (Balcan et al., 2007), which we will refer to as the *preference-based setting*. In the first stage of that setting, a preference function $h : U \times U \mapsto [0, 1]$ is learned, where values of $h(u, v)$ closer to one indicate that u is ranked above v and values closer to zero the opposite. h is typically assumed to be the output of a classification algorithm trained on a sample of labeled pairs, and can be for example a convex combination of simpler preference functions as in (Cohen et al., 1999). A crucial difference with the score-based setting is that, in general, the preference function h may not induce a linear ordering. The relation it induces may be non-transitive, thus we may have for example $h(u, v) = h(v, w) = h(w, u) = 1$ for three distinct points u, v, and w. To rank a test subset $V \subseteq U$, in the second stage, the algorithm orders the points in V by making use of the preference function h learned in the first stage. The subset ranking setup examined by Cossock and Zhang (2006), though distinct, also bears some resemblance with this setting.

This paper deals with the preference-based ranking setting just described. The advantage of this setting is that the learning algorithm is not required to return a linear ordering of all points in U, which may be impossible to achieve faultlessly in accordance with a general possibly non-transitive pairwise preference labeling. This is more likely to be achievable exactly or with a better approximation when the algorithm is requested instead, to supply a linear ordering, only for limited subsets $V \subseteq U$.

When the preference function is obtained as the output of a binary classification algorithm, the preference-based setting can be viewed as a reduction of ranking to classification. The second stage specifies how the ranking is obtained using the preference function.

Cohen et al. (1999) showed that in the second stage of the preference-based setting, the general problem of finding a linear ordering with as few pairwise misrankings as possible with respect to the preference function h is NP-complete. The authors presented a greedy algorithm based on the tour-

nament degree, that is, for a given element u, the difference between the number of elements it is preferred to versus the number of those preferred to u. The bound proven by the authors, formulated in terms of the pairwise disagreement loss l with respect to the preference function h, can be written as $l(\sigma_{greedy}, h) \leq 1/2 + l(\sigma_{optimal}, h)/2$, where $l(\sigma_{greedy}, h)$ is the loss achieved by the permutation σ_{greedy} returned by their algorithm and $l(\sigma_{optimal}, h)$ the one achieved by the optimal permutation $\sigma_{optimal}$ with respect to the preference function h. This bound was given for the general case of ranking, but, in the particular case of bipartite ranking, a random ordering can achieve a pairwise disagreement loss of $1/2$ and thus the bound is not informative. Note that the algorithm can be viewed as a derandomization technique.

More recently, Balcan et al. (2007) studied the bipartite ranking problem. In this particular case, the loss of an output ranking is measured by counting pairs of ranked elements, one of which is positive and the other negative (based on some ground truth). They showed that sorting the elements of V according to the same tournament degree used by Cohen et al. (1999) guarantees a regret of at most $2r$ using a binary classifier with regret r. (The regret is defined as a calibration of the loss function that aligns a theoretical optimum with 0.) However, due to the quadratic nature of the definition of the tournament degree, their algorithm requires $\Omega(n^2)$ calls to the preference function h, where $n = |V|$ is the number of objects to rank.

We describe an efficient randomized algorithm for the second stage of preference-based setting and thus for reducing the learning problem of ranking to binary classification. We improve on the recent result of Balcan et al. (2007), by guaranteeing a pairwise misranking regret of at most r using a binary classifier with regret r, thereby improving the bound by a factor of 2. Our reduction applies, with different constants, to a broader class of ranking loss functions, admits a simple proof, and the expected running time complexity of our algorithm in terms of number of calls to a classifier or preference function is improved from $\Omega(n^2)$ to $O(n \log n)$. Furthermore, when the top k ranked elements only are required ($k \ll n$), as in many applications in information extraction or search engines, the time complexity of our algorithm can be further reduced to $O(k \log k + n)$. Our reduction and algorithm are thus practical for realistic applications where the number of points to rank exceeds several thousands. The price paid for this improvement is in resorting to nondeterminism. Indeed, our algorithms are randomized, but this turns our to be necessary. We give a simple proof of a lower bound of $2r$ for *any* deterministic reduction of ranking to binary classification with classification regret r, thereby generalizing to all deterministic reductions a lower bound result of Balcan et al. (2007).

To appreciate our improvement of the reduction bound from a factor of 2 to 1, consider the case of a binary classifier with an error rate of just 25%, which is quite reasonable in many applications. Assume that the Bayes error is close to zero for the classification problem and similarly that for the ranking problem that the regret and loss approximately coincide. Then, the bound of Balcan et al. (2007) guarantees for the ranking algorithm a pairwise misranking error of at most 50%. But, since a random ranking can achieve 50% pairwise misranking error, the bound turns out not to be informative in that case. Instead, with a factor of 1, the bound ensures a pairwise misranking of at most 25%.

Much of our results also extend beyond the bipartite case previously studied by Balcan et al. (2007) to the general case of ranking. A by-product of our proofs is a bound on the pairwise disagreement loss with respect to the preference function h that we will compare to the result given by Cohen et al. (1999).

The algorithm used by Balcan et al. (2007) to produce a ranking based on the preference function is known as sort-by-degree and has been recently used in the context of minimizing the feedback arcset in tournaments (Coppersmith et al., 2006). Here, we use a different algorithm, QuickSort, which has also been recently used for minimizing the feedback arcset in tournaments (Ailon et al. 2005; 2007). The techniques presented build upon earlier work by Ailon et al. (2005; 2007) on combinatorial optimization problems over rankings and clustering.

The remainder of the paper is structured as follows. In Section 2, we introduce the definitions and notation used in future sections and introduce a general family of loss functions for ranking. Section 3 describes a simple and efficient algorithm for reducing ranking to binary classification, proves several bounds guaranteeing the quality of the ranking produced by the algorithm, and analyzes the running-time complexity of our algorithm. In Section 4, we derive a lower bound for any deterministic reduction of ranking to binary classification. In Section 5, we discuss the relationship of the algorithm and its proof with previous related work in combinatorial optimization, and discuss key assumptions related to the notion of regret in this context.

2 Preliminaries

This section introduces several preliminary definitions necessary for the presentation of our results. In what follows, U will denote a universe of elements, e.g., the collection of all possible query-result pairs returned by a web search task, and $V \subseteq U$ will denote a small subset thereof, e.g., a preliminary list of relevant results for a given query. For simplicity of notation we will assume that U is a set of integers, so that we are always able to choose a minimal canonical element in a finite subset, as we do in (9) below. This arbitrary ordering should not be confused with the ranking problem we are considering.

2.1 General Definitions and Notation

We first briefly discuss the learning setting and assumptions made here and compare them with those of Balcan et al. (2007) and Cohen et al. (1999).

In what follows, $V \subseteq U$ represents a finite subset extracted from some arbitrary universe U, which is the set we wish to rank at each round. The notation $S(V)$ denotes the set of *rankings* on V, that is the set of injections from V to $[n] = \{1, \ldots, n\}$, where $n = |V|$. If $\sigma \in S(V)$ is such a ranking, then $\sigma(u)$ is the rank of an element $u \in V$, where lower ranks are interpreted as preferable ones. More precisely, we say that u is preferred over v with respect to σ if $\sigma(u) < \sigma(v)$. For convenience, and abusing notation, we

also write $\sigma(u,v) = 1$ if $\sigma(u) < \sigma(v)$ and $\sigma(u,v) = 0$ otherwise. We let $\binom{V}{k}$ denote the collection of all subsets of size exactly k of V. To distinguish between functions taking ordered vs. unordered arguments in what follows, we will use the notation $F_{u_1 u_2 \ldots u_k}$ to denote k unordered arguments for a function F defined on $\binom{V}{k}$ and $F(u_1, u_2, \ldots, u_k)$ to denote k ordered arguments for a function F defined on $\underbrace{V \times \cdots \times V}_{k}$.

2.2 Ground truth

As in standard learning scenarios, at each round, there is an underlying unknown ground truth which we wish the output of the learning algorithm to agree with as much as possible. The ground truth is a ranking that we denote by $\sigma^* \in S(V)$, equipped with a function ω assigning different *importance* weight to pairs of positions. The combination (σ^*, ω) is extremely expressive, as we shall see below in Section 2.5. It can encode in particular the standard average pairwise misranking or AUC loss assumed by Balcan et al. (2007) in a bipartite setting, but also more sophisticated ones capturing misrankings among the top k, and other losses that are close but distinct from those considered by Clémençon and Vayatis (2007).

2.3 Preference function

As with both (Cohen et al., 1999) and (Balcan et al., 2007), we assume that a preference function $h\colon U \times U \to [0,1]$ is learned in a first learning stage. The convention is that the higher $h(u,v)$ is, the more our belief that u should be preferred to v. The function h satisfies *pairwise consistency*: $h(u,v) + h(v,u) = 1$, but need not even be transitive on 3-tuples (cycles may be induced). The second stage uses h to output a proper ranking σ, as we shall further discuss below. The running time complexity of the second stage is measured with respect to the number of calls to h.

2.4 Output of Learning Algorithm

The final output of the second stage of the algorithm, σ, is a proper ranking of V. Its cost is measured differently in (Balcan et al., 2007) and (Cohen et al., 1999). In the former, it is measured against the unknown ground truth and compared to the cost of h against the ground truth. The rationale is that the information encoded in h contains all pairwise preference information using the state-of-the-art binary classification. In (Cohen et al., 1999), σ is measured against the given preference function h, and compared to the theoretically best one can obtain. Thus, there h plays the role of a known ground truth.

2.5 Loss Functions

We are now ready to define the loss functions used to measure the quality of an output ranking σ either with respect to σ^*, as in (Balcan et al., 2007), or with respect to h, as in (Cohen et al., 1999).

The following general loss function L_ω measures the quality of a ranking σ with respect to a desired one σ^* using a weight function ω (described below):

$$L_\omega(\sigma, \sigma^*) = \binom{n}{2}^{-1} \sum_{u \neq v} \sigma(u,v)\sigma^*(v,u)\omega(\sigma^*(u), \sigma^*(v)).$$

The sum is over all pairs u, v in the domain V of the rankings σ, σ^*. It counts the number of inverted pairs $u, v \in V$ weighed by ω, which assigns importance coefficients to pairs, based on their positions in the ground truth σ^*. The function ω must satisfy the following three natural axioms, which will be necessary in our analysis:

(P1) Symmetry: $\omega(i,j) = \omega(j,i)$ for all i, j;

(P2) Monotonicity: $\omega(i,j) \leq \omega(i,k)$ if either $i < j < k$ or $i > j > k$;

(P3) Triangle inequality: $\omega(i,j) \leq \omega(i,k) + \omega(k,j)$.

This definition is very general and encompasses many useful, well studied distance functions. Setting $\omega(i,j) = 1$ for all $i \neq j$ yields the unweighted pairwise misranking measure or the so-called Kemeny distance function.

For a fixed integer k, the following function

$$\omega(i,j) = \begin{cases} 1 & \text{if } ((i \leq k) \vee (j \leq k)) \wedge (i \neq j) \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

can be used to emphasize ranking at the top k elements. Misranking of pairs with one element ranked among the top k is penalized by this function. This can be of interest in applications such as information extraction or search engines where the ranking of the top documents matters more. For this emphasis function, all elements ranked below k are in a tie. In fact, it is possible to encode any tie relation using ω.

Bipartite Ranking. In a bipartite ranking scenario, V is partitioned into a positive and negative set V^+ and V^- of sizes m^+ and m^- respectively, where $m^+ + m^- = |V| = n$. For this scenario (Balcan et al., 2007; Hanley & McNeil, 1982; Lehmann, 1975), we are often interested in the AUC score of $\sigma \in S(V)$ defined as follows:

$$1 - \text{AUC}(V^+, V^-, \sigma) = \frac{1}{m^- m^+} \sum_{u,v \in V} \mathbf{1}_{(u,v) \in V^+ \times V^-} \sigma(v,u).$$

This expression measures the probability given a random *crucial* pair of elements, one of which is positive and the other negative, that the pair is misordered in σ. It is immediate to verify that this is equal to $L_\omega(\sigma, \sigma^*)$, where σ^* is any ranking placing V^+ ahead of V^-, and

$$\omega(i,j) = \frac{\binom{n}{2}}{m^- m^+} \begin{cases} 1 & (i \leq m^+) \wedge (j > m^+) \\ 1 & (j \leq m^+) \wedge (i > m^+) \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Simplified notation. To avoid carrying σ^* and ω, we will define for convenience

$$\tau^*(u,v) = \sigma^*(u,v)\omega(\sigma^*(u), \sigma^*(v))$$

and

$$L(\sigma, \tau^*) := L_\omega(\sigma, \sigma^*) = \binom{n}{2}^{-1} \sum_{u \neq v} \sigma(u,v)\tau^*(v,u) \ .$$

We will formally call τ^* a *generalized ranking*, and it will take the role of the ground truth. If ω is obtained as in (2) for some integers m^+, m^- satisfying $m^+ + m^- = n$ then we will say that the corresponding τ^* is *bipartite*.

It is immediate to verify from the properties of the weight function ω that for all $u, v, w \in V$,

$$\tau^*(u,v) \leq \tau^*(u,w) + \tau^*(w,v) \ . \tag{3}$$

If τ^* is bipartite, then additionally,

$$\tau^*(u,v) + \tau^*(v,w) + \tau^*(w,u) = \tau^*(v,u) + \tau^*(w,v) + \tau^*(u,w) \ . \tag{4}$$

2.6 Preference Loss Function

We need to extend the definition to measure the loss of a preference function h with respect to σ^*. In contrast with the loss function just defined, we need to define a *preference loss* measuring a generalized ranking's disagreements with respect to a preference function h when measured against τ^*. We can readily extend the loss definitions defined above as follows:

$$L(h,\tau^*) = L_\omega(h,\sigma^*) = \binom{n}{2}^{-1} \sum_{u \neq v} h(u,v)\tau^*(v,u) \ .$$

As explained above, $L(h,\tau^*)$ is the ideal loss the learning algorithm will aim to achieve with the output ranking hypothesis σ.

2.7 Input Distribution

The set V we wish to rank together with the ground truth τ^* are drawn as pair from a distribution we denote by D. In other words, τ^* may be a random function of V. For our analysis of the loss though, it is convenient to think of V and τ^* as fixed, because our bounds will be conditioned on fixed V, τ^* and will easily generalize to the stochastic setting. Finally, we say that D is bipartite if τ^* is bipartite with probability 1.

2.8 Regret Functions

The notion of regret is commonly used to measure the difference between the loss incurred by a learning algorithm and that of some *best* alternative. This section introduces the definitions of regret that we will be using to quantify the quality of a ranking algorithm in this context. We will define a notion of *weak* and *strong* regret for both ranking and classification losses as follows.

To define a strong ranking regret, we subtract from the loss function the minimal loss that could have been obtained from a global ranking $\tilde{\sigma}$ of U. More precisely, we define:

$$\mathcal{R}_{rank}(A,D) = E_{V,\tau^*,s}[L(A_s(V),\tau^*)] - \min_{\tilde{\sigma} \in S(U)} E_{V,\tau^*}[L(\tilde{\sigma}_{|V},\tau^*)] \ ,$$

where $\tilde{\sigma}_{|V} \in S(V)$ is defined by restricting the ranking $\tilde{\sigma} \in S(U)$ to V in a natural way, and A is a possibly randomized algorithm using a stream of random bits s (and a pre-learned preference function h) to output a ranking $A_s(V)$ in $S(V)$.

As for the strong preference loss, it is natural to subtract the minimal loss over all, possibly cyclic, preference functions on U.

More precisely, we define:

$$\mathcal{R}_{class}(h,D) = E_{V,\tau^*}[L(h_{|V},\tau^*)] - \min_{\tilde{h}} E_{V,\tau^*}[L(\tilde{h}_{|V},\tau^*)] \ ,$$

where the minimum is over $\tilde{h}$, a preference function over U, and $\cdot_{|V}$ is a restriction operator on preference functions defined in the natural way.

The weak ranking and classification regret functions $\mathcal{R}'_{rank}$ and $\mathcal{R}'_{class}$ are defined as follows:

$$\mathcal{R}'_{rank}(A,D) = E_{V,\tau^*,s}[L(A_s(V),\tau^*)] - E_V \min_{\tilde{\sigma} \in S(V)} E_{\tau^*|V}[L(\tilde{\sigma},\tau^*)] \tag{5}$$

$$\mathcal{R}'_{class}(h,D) = E_{V,\tau^*}[L(h_{|V},\tau^*)] - E_V \min_{\tilde{h}} E_{\tau^*|V}[L(\tilde{h},\tau^*)] \ , \tag{6}$$

where $\tau^*|V$ is the random variable τ^* conditioned on fixed V. The difference between $\mathcal{R}$ and $\mathcal{R}'$ for both ranking and classification is that in their definition the $\min$ operator and the E_V operator are permuted.

The following inequalities follow from the concavity of $\min$ and Jensen's inequality:

$$\begin{aligned} &\mathcal{R}'_{rank}(A,D) \geq \mathcal{R}_{rank}(A,D) \quad \text{and} \\ &\mathcal{R}'_{class}(A,D) \geq \mathcal{R}_{class}(A,D). \end{aligned} \tag{7}$$

For a fixed V and any $u, v \in V$, let

$$e(u,v) = E_{\tau^*|V}[\tau^*(u,v)] \ . \tag{8}$$

The reason we work with $\mathcal{R}'_{class}$ is because the preference function $\tilde{h}$ over U obtaining the $\min$ in the definition of $\mathcal{R}'_{class}$ can be determined locally for any $u, v \in U$ by

$$\tilde{h}(u,v) = \begin{cases} 1 & e(u,v) > e(v,u) \\ 0 & e(v,u) > e(u,v) \\ \mathbf{1}_{u>v} & \text{otherwise} \ . \end{cases} \tag{9}$$

Also, equation (3) holds true with e replacing τ^*, and similarly for (4) if D is bipartite (by linearity of expectation). We cannot do a similar thing when working with the strong regret function $\mathcal{R}_{class}$.

The reason we work with weak ranking regret is for compatibility with our choice of weak classification regret, although our upper bounds on $\mathcal{R}'_{rank}$ trivially apply to $\mathcal{R}_{rank}$ in virtue of (7).

In Section 5.4, we will discuss certain assumptions under which our results work for the notion of strong regret as well. Note that Balcan et al.(2007) also implicitly use such an assumption in deriving their regret bounds. Our regret bounds (second part of Theorem 2) hold under the same assumption. Our result is thus exactly comparable with theirs.

3 Algorithm for Ranking Using a Preference Function

This section describes and analyzes an algorithm for obtaining a global ranking of a subset using a prelearned preference function h, which corresponds to the second stage of the preference-based setting. Our bound on the loss will be derived using conditional expectation on the preference loss assuming a fixed subset $V \subseteq U$, and fixed ground truth τ^*.

To further simplify the analysis, we assume that h is binary, that is $h(u,v) \in \{0,1\}$ for all $u, v \in U$.

3.1 Description

One simple idea to obtain a global ranking of the points in V consists of using a standard comparison-based sorting algorithm where the comparison operation is based on the preference function. However, since in general the preference function is not transitive, the property of the resulting permutation obtained is unclear.

This section shows however that the permutation generated by the standard QuickSort algorithm provides excellent guarantees.[1] Thus, the algorithm we suggest is the following. Pick a random *pivot* element u uniformly at random from V. For each $v \neq u$, place v on the left[2] of u if $h(v,u) = 1$, and to its right otherwise. Proceed recursively with the array to the left of u and the one to its right and return the concatenation of the permutation returned by the left recursion, u, and the permutation returned by the right recursion.

We will denote by $Q_s^h(V)$ the permutation resulting in running QuickSort on V using preference function h, where s is the random stream of bits used by QuickSort for the selection of the pivots. As we shall see in the next two sections, this algorithm produces high-quality global rankings in a time-efficient manner.

3.2 Ranking Quality Guarantees

The following theorems bound the ranking quality of the algorithm described, for both loss and regret, in the general and bipartite cases.

Theorem 1 (Loss bounds in general case) *For any fixed subset $V \subseteq U$, preference function h on V, and generalized ranking τ^* on V, the following bound holds:*

$$\mathop{\mathrm{E}}_{s}[L(Q_s^h(V), \tau^*)] \leq 2L(h, \tau^*) \ . \tag{10}$$

Taking the expectation of both sides, this implies immediately that

$$\mathop{\mathrm{E}}_{V,\tau^*,s}[L(Q_s^h(V), \tau^*)] \leq 2E_{V,\tau^*}[L(h, \tau^*)], \tag{11}$$

where h could depend on V.

Theorem 2 (Loss and regret bounds in bipartite case) *For any fixed $V \subseteq U$, preference function h over V, and bipartite generalized ranking τ^*, the following bound holds:*

$$\mathop{\mathrm{E}}_{s}[L(Q_s^h(V), \tau^*] = L(h, \tau^*) \tag{12}$$

$$\mathcal{R}'_{rank}(Q_s^h(\cdot), D) \leq \mathcal{R}'_{class}(h, D) \ . \tag{13}$$

Taking the expectation of both sides of Equation 12, this implies immediately that if (V, τ^*) is drawn from a bipartite distribution D, then

$$\mathop{\mathrm{E}}_{V,\tau^*,s}[L(Q_s^h(V), \tau^*)] = E_{V,\tau^*}[L(h, \tau^*)], \tag{14}$$

where h can depend on V.

To present the proof of these theorems, we need some tools helpful in the analysis of QuickSort, similar to those originally developed by Ailon et al.(2005). The next section introduces these tools.

[1] We are not assuming here transitivity as in standard textbook presentations of QuickSort.

[2] We will use the convention that ranked items are written from left to right, starting with the most preferred ones.

3.3 Analysis of QuickSort

Assume V is fixed, and let $Q_s = Q_s^h(V)$ be the (random) ranking output by QuickSort on V using the preference function h. During the execution of QuickSort, the order between two elements $u, v \in V$ is determined in one of two ways:

- Directly: u (or v) was selected as the pivot with v (resp. u) present in the same sub-array in a recursive call to QuickSort. We denote by $p_{uv} = p_{vu}$ the probability of that event. In that case, the algorithm orders u and v according to the preference function h.
- Indirectly: a third element $w \in V$ is selected as pivot with w, u, v all present in the same sub-array in a recursive call to QuickSort, u is assigned to the left sub-array and v to the right (or vice-versa).

  Let p_{uvw} denote the probability of the event that u, v, and w are present in the same array in a recursive call to QuickSort and that one of them is selected as pivot. Note that conditioned on that event, each of these three elements is equally likely to be selected as a pivot since the pivot selection is based on a uniform distribution.

  If (say) w is selected among the three, then u will be placed on the left of v if $h(u,w) = h(w,v) = 1$, and to its right if $h(v,w) = h(w,u) = 1$. In all other cases, the order between u, v will be determined only in a deeper nested call to QuickSort.

Let $X, Y \colon V \times V \to \mathbb{R}$ be any two functions on ordered pairs $u, v \in V$, and let $Z \colon \binom{V}{2} \to \mathbb{R}$ be a function on unordered pairs. We define three functions $\alpha[X,Y] \colon \binom{V}{2} \to \mathbb{R}$, $\beta[X] \colon \binom{V}{3} \to \mathbb{R}$ and $\gamma[Z] \colon \binom{V}{3} \to \mathbb{R}$ as follows:

$$\alpha[X,Y]_{uv} = X(u,v)Y(v,u) + X(v,u)Y(u,v),$$

$$\beta[X]_{uvw} =$$
$$\frac{1}{3}(h(u,v)h(v,w)X(w,u) + h(w,v)h(v,u)X(u,w)) +$$
$$\frac{1}{3}(h(v,u)h(u,w)X(w,v) + h(w,u)h(u,v)X(v,w)) +$$
$$\frac{1}{3}(h(u,w)h(w,v)X(v,u) + h(v,w)h(w,u)X(u,v)),$$

$$\gamma[Z]_{uvw} =$$
$$\frac{1}{3}(h(u,v)h(v,w) + h(w,v)h(v,u))Z_{uw} +$$
$$\frac{1}{3}(h(v,u)h(u,w) + h(w,u)h(u,v))Z_{vw} +$$
$$\frac{1}{3}(h(u,w)h(w,v) + h(v,w)h(w,u))Z_{uv} \ .$$

Lemma 3 (QuickSort Decomposition)

1. *For any $Z \colon \binom{V}{2} \to \mathbb{R}$,*
$$\sum_{u<v} Z_{uv} = \sum_{u<v} p_{uv} Z_{uv} + \sum_{u<v<w} p_{uvw}\gamma[Z]_{uvw} \ .$$
2. *For any $X \colon V \times V \to \mathbb{R}$,*
$$E_s[\sum_{u<v} \alpha[Q_s, X]_{uv}] =$$
$$\sum_{u<v} p_{uv}\alpha[h,X]_{uv} + \sum_{u<v<w} p_{uvw}\beta[X]_{uvw} \ .$$

Proof: To see the first part, notice that for every unordered pair $u < v$ the expression Z_{uv} is accounted for on the RHS of the equation with total coefficient:

$$p_{uv} + \sum_{w \notin \{u,v\}} \frac{1}{3} p_{uvw}(h(u,w)h(w,v) + h(v,w)h(w,u)) \ .$$

Now, p_{uv} is the probability that the order of (u, v) is determined directly (by definition), and

$$\frac{1}{3} p_{uvw}(h(u,w)h(w,v) + h(v,w)h(w,u))$$

is the probability that their order is determined indirectly via w as pivot. Since each pair's ordering is accounted for exactly once, these probabilities are for pairwise disjoint events that cover the probability space. Thus, the total coefficient of Z_{uv} on the RHS is 1, as is on the LHS. The second part is proved similarly. ■

3.4 Loss Bounds

This section proves Theorem 1 and the first part of Theorem 2. For a fixed τ^*, the loss incurred by QuickSort is $L(Q_s, \tau^*) = \binom{n}{2}^{-1} \sum_{u<v} \alpha[Q_s, \tau^*]_{uv}$. By the second part of Lemma 3, the expected loss is therefore

$$\mathop{\mathrm{E}}_{s}[L(Q_s, \tau^*)] = \binom{n}{2}^{-1} \left(\sum_{u<v} p_{uv} \alpha[h, \tau^*]_{uv} + \sum_{u<v<w} p_{uvw} \beta[\tau^*]_{uvw} \right) \ .$$

Also, the following holds by definition of L:

$$L(h, \tau^*) = \binom{n}{2}^{-1} \sum_{u<v} \alpha[h, \tau^*]_{uv} \ .$$

Thus, by the first part of Lemma 3,

$$L(h, \tau^*) = \binom{n}{2}^{-1} \left(\sum_{u<v} p_{uv} \alpha[h, \tau^*]_{uv} + \sum_{u<v<w} \gamma[\alpha[h, \tau^*]]_{uvw} \right) \ .$$

To complete the proof, it suffices to show that for all u, v, w,

$$\beta[\tau^*]_{uvw} \le 2\gamma[\alpha[h, \tau^*]]_{uvw} \ , \tag{15}$$

and that if τ^* is bipartite, then

$$\beta[\tau^*]_{uvw} = \gamma[\alpha[h, \tau^*]]_{uvw} \ . \tag{16}$$

Up to symmetry, there are two cases to consider. The first case assumes that h induces a cycle on u, v, w, the second assumes that it doesn't.

1. Without loss of generality, assume $h(u, v) = h(v, w) = h(w, u) = 1$. Plugging in the definitions leads to

$$\beta[\tau^*]_{uvw} = \frac{1}{3}(\tau^*(u,v) + \tau^*(v,w) + \tau^*(w,u)), \text{ and} \tag{17}$$

$$\gamma[\alpha[h, \tau^*]]_{uvw} = \frac{1}{3}(\tau^*(v,u) + \tau^*(w,v) + \tau^*(u,w)) \ . \tag{18}$$

   If τ^* is bipartite, then by (4) the right hand sides of (17) and (18) are equal, giving (16). Otherwise we use (3) to derive

$$\begin{aligned} \tau^*(u,v) &\le \tau^*(u,w) + \tau^*(w,v) \\ \tau^*(v,w) &\le \tau^*(v,u) + \tau^*(u,w) \\ \tau^*(w,u) &\le \tau^*(w,v) + \tau^*(v,u) \end{aligned}$$

   Summing up the three equations, this implies (15).

2. Without loss of generality, assume $h(u, v) = h(v, w) = h(u, w) = 1$. Plugging in the definitions gives

$$\beta[\tau^*]_{uvw} = \gamma[\alpha[h, \tau^*]]_{uvw} = \tau^*(w, u)$$

   as required. ■

We now examine a consequence of Theorem 1 for QuickSort that can be compared with the bound given by Cohen et al. (1999) for a greedy algorithm based on the tournament degree. Let $\sigma_{optimal}$ be the ranking with the least amount of pairwise disagreement with h:

$$\sigma_{optimal} = \operatorname*{argmin}_{\sigma} L(h, \sigma) \ .$$

Then, the following corollary bounds the expected pairwise disagreement of QuickSort with respect to $\sigma_{optimal}$ by twice that of the preference function with respect to $\sigma_{optimal}$.

Corollary 4 *For any $V \subseteq U$ and preference function h over V, the following bound holds:*

$$\mathop{\mathrm{E}}_{s}[L(Q^h_s(V), \sigma_{optimal})] \le 2\, L(h, \sigma_{optimal}) \ . \tag{19}$$

The corollary is immediate since technically any ranking, in particular $\sigma_{optimal}$, can be taken as σ^* in the proof of Theorem 1.

Corollary 5 *Let $V \subseteq U$ be an arbitrary subset of U and let $\sigma_{optimal}$ be as above. Then, the following bound holds for the pairwise disagreement of the ranking $Q^h_s(V)$ with respect to h:*

$$\mathop{\mathrm{E}}_{s}[L(h, Q^h_s(V))] \le 3\, L(h, \sigma_{optimal}). \tag{20}$$

Proof: The result follows directly Corollary 4 and the application of the triangle inequality. ■

This result is in fact known from previous work (Ailon et al. 2005; 2007) where it is proven directly without resorting to the intermediate inequality (19). In fact, a better factor of 2.5 is known to be achievable using a more complicated algorithm, which gives hope for a 1.5 bound improving Theorem 1.

3.5 Regret Bounds for Bipartite case

This section proves the second part of Theorem 2, that is the regret bound. Since in the definition of $\mathcal{R}'_{rank}$ and $\mathcal{R}'_{class}$ the expectation over V is outside the $\min$ operator, we may continue to fix V. Let D_V denote the distribution over the bipartite τ^* conditioned on V. By the definitions of $\mathcal{R}'_{rank}$ and $\mathcal{R}'_{class}$, it is now sufficient to prove that

$$\mathop{\mathrm{E}}_{\tau^*|V,s}[L(Q^h_s, \tau^*)] - \min_{\tilde{\sigma}} \mathop{\mathrm{E}}_{\tau^*|V}[L(\tilde{\sigma}, \tau^*)] \le \mathop{\mathrm{E}}_{\tau^*|V}[L(h, \tau^*)] - \min_{\tilde{h}} \mathop{\mathrm{E}}_{\tau^*|V}[L(\tilde{h}, \tau^*)]. \tag{21}$$

We let $e(u,v)$ denote $\mathrm{E}_{\tau^*|V}[\tau^*(u,v)]$, then by the linearity of expectation, $\mathrm{E}_{\tau^*|V}[L(\tilde{\sigma},\tau^*)] = L(\tilde{\sigma},e)$ and similarly $\mathrm{E}_{\tau^*|V}[L(\tilde{h},\tau^*)] = L(\tilde{h},e)$. Thus, inequality 21 can be rewritten as

$$\mathop{\mathrm{E}}_{s}[L(Q_s^h,e)] - \min_{\tilde{\sigma}} L(\tilde{\sigma},e) \le L(h,e) - \min_{\tilde{h}} L(\tilde{h},e). \quad (22)$$

Now let $\tilde{\sigma}$ and $\tilde{h}$ be the minimizers of the $\min$ operators on the left and right sides, respectively. Recall that for all $u,v \in V$, $\tilde{h}(u,v)$ can be taken greedily as a function of $e(u,v)$ and $e(v,u)$, as in (9):

$$\tilde{h}(u,v) = \begin{cases} 1 & e(u,v) > e(v,u) \\ 0 & e(u,v) < e(v,u) \\ \mathbf{1}_{u>v} & \text{otherwise (equality)} \ . \end{cases} \quad (23)$$

Using Lemma 3 and linearity, the LHS of (22) can be rewritten as:

$$\binom{n}{2}^{-1} \left(\sum_{u<v} p_{uv}\ \alpha[h-\tilde{\sigma},e]_{uv} + \sum_{u<v<w} p_{uvw}(\beta[e] - \gamma[\alpha[\tilde{\sigma},e]])_{uvw} \right) \ ,$$

and the RHS of (22) as:

$$\binom{n}{2}^{-1} \left(\sum_{u<v} p_{uv}\ \alpha[h-\tilde{h},e]_{uv} + \sum_{u<v<w} p_{uvw}\gamma[\alpha[h-\tilde{h},e]]_{uvw} \right) \ .$$

Now, clearly, for all (u,v) by construction of $\tilde{h}$, we must have $\alpha[h-\tilde{\sigma},e]_{uv} \le \alpha[h-\tilde{h},e]_{uv}$. To conclude the proof of the theorem, we define $F\colon \binom{n}{3} \to \mathbb{R}$ as follows:

$$F = \beta[e] - \gamma[\alpha[\tilde{\sigma},e]] - (\gamma[\alpha[h,e]] - \gamma[\alpha[\tilde{h},e]]) \ . \quad (24)$$

It now suffices to prove that $F_{uvw} \le 0$ for all $u,v,w \in V$. Clearly F is a function of the values of

$$\begin{aligned} e(a,b) &: \{a,b\} \subseteq \{u,v,w\} \\ h(a,b) &: \{a,b\} \subseteq \{u,v,w\} \\ \tilde{\sigma}(a,b) &: \{a,b\} \subseteq \{u,v,w\}. \end{aligned} \quad (25)$$

Recall that $\tilde{h}$ depends on e. By (3) and (4), the e-variables can take values satisfying the following constraints for all $u,v,w \in V$:

$$\forall \{a,b,c\} = \{u,v,w\}\,,\ e(a,c) \le e(a,b) + e(b,c) \quad (26)$$

$$e(u,v) + e(v,w) + e(w,u) = e(v,u) + e(w,v) + e(u,w) \quad (27)$$

$$\forall a,b \in \{u,v,w\}\,,\ e(a,b) \ge 0 \ . \quad (28)$$

Let $P \subseteq \mathbb{R}^6$ denote the polytope defined by (26-28) in the variables $e(a,b)$ for $\{a,b\} \subseteq \{u,v,w\}$. We subdivide P into smaller subpolytopes on which the $\tilde{h}$ variables are constant. Up to symmetries, we can consider only two cases: (i) $\tilde{h}$ induces a cycle on u,v,w and (ii) $\tilde{h}$ is cycle-free on u,v,w.

(i) Without loss of generality, assume $\tilde{h}(u,v) = \tilde{h}(v,w) = \tilde{h}(w,u) = 1$. But this implies that $e(u,v) \ge e(v,u)$, $e(v,w) \ge e(w,v)$ and $e(w,u) \ge e(u,w)$. Together with (27) and (28), this implies that $e(u,v) = e(v,u)$, $e(v,w) = e(w,v)$, and $e(w,u) = e(u,w)$. Consequently,

$$\begin{aligned} \beta[e]_{uvw} &= \gamma[\alpha[\tilde{\sigma},e]]_{uvw} \\ &= \gamma[\alpha[h,e]]_{uvw} = \gamma[\alpha[\tilde{h},e]]_{uvw} \\ &= \frac{1}{3}(e(u,v) + e(v,w) + e(w,u)) \ , \end{aligned}$$

and $F_{uvw} = 0$, as required.

(ii) Without loss of generality, assume $\tilde{h}(u,v) = \tilde{h}(v,w) = \tilde{h}(u,w) = 1$. This implies that

$$\begin{aligned} e(u,v) &\ge e(v,u) \\ e(v,w) &\ge e(w,v) \\ e(u,w) &\ge e(w,u) \ . \end{aligned} \quad (29)$$

Let $\tilde{P} \subseteq P$ denote the polytope defined by (29) and (26)-(28). Clearly, F is linear in the 6 e variables when all the other variables are fixed. Since F is also homogenous in the e variables, it suffices to prove that $F \le 0$ for e taking values in $\tilde{P}' \subseteq \tilde{P}$, which is defined by adding the constraint, say,

$$\sum_{a,b \in \{u,v,w\}} e(a,b) = 2 \ .$$

It is now enough to prove that $F \le 0$ for τ^* being a vertex of of $\tilde{P}'$. This finite set of cases can be easily checked to be:

$$\begin{aligned} &(e(u,v), e(v,u), e(u,w), \\ &\qquad e(w,u), e(w,v), e(v,w)) \in A \cup B \ , \end{aligned}$$

where

$$\begin{aligned} A =& \{(0,0,1,0,0,1), (1,0,1,0,0,0)\} \\ B =& \{(.5,.5,.5,.5,0,0), (.5,.5,0,0,.5,.5), \\ & (0,0,.5,.5,.5,.5)\} \ . \end{aligned}$$

The points in B were already checked in case (i), which is, geometrically, a boundary of case (ii). It remains to check the two points in A.

- case $(0,0,1,0,0,1)$: plugging in the definitions, one checks that:

  $$\begin{aligned} &\beta[e]_{uvw} = \frac{1}{3}(h(w,v)h(v,u) + h(w,u)h(u,v)) \\ &\gamma[\alpha[h,e]]_{uvw} = \\ &\quad \frac{1}{3}((h(u,v)h(v,w) + h(w,v)h(v,u))h(w,u) \\ &\qquad + (h(v,u)h(u,w) + h(w,u)h(u,v))h(w,v)) \\ &\gamma[\alpha[\tilde{h},e]]_{uvw} = 0 \ . \end{aligned}$$

  Clearly F could be positive only of $\beta_{uvw} = 1$, which happens if and only if either $h(w,v)h(v,u) =$

1 or $h(w,u)h(u,v) = 1$. In the former case, we obtain that

$$\text{either} \quad h(w,v)h(v,u)h(w,u) = 1 \tag{30}$$
$$\text{or} \quad h(v,u)h(u,w)h(w,v) = 1 \ , \tag{31}$$

both implying that $\gamma[\alpha[h,e]]_{uvw} \geq 1$, thus $F \leq 0$. In the latter case,

$$\text{either} \quad h(w,u)h(u,v)h(w,v) = 1 \tag{32}$$
$$\text{or} \quad h(u,v)h(v,w)h(w,u) = 1 \ , \tag{33}$$

both implying again that $\gamma[\alpha[h,e]]_{uvw} \geq 1$ and thus $F \leq 0$.

- case $(1,0,1,0,0,0)$: plugging in the definitions, one checks that:

$$\beta[e]_{uvw} = \frac{1}{3}(h(w,v)h(v,u) + h(v,w)h(w,u))$$
$$\gamma[\alpha[h,e]]_{uvw} =$$
$$\frac{1}{3}((h(u,v)h(v,w) + h(w,v)h(v,u))h(w,u)$$
$$+ (h(u,w)h(w,v) + h(v,w)h(w,u))h(v,u)) \ .$$
$$\gamma[\alpha[\tilde{h},e]]_{uvw} = 0 \ .$$

Now F could be positive if and only if

$$\text{either} \quad h(w,v)h(v,u) = 1 \tag{34}$$
$$\text{or} \quad h(v,w)h(w,u) = 1 \ . \tag{35}$$

In the former case, we obtain that

$$\text{either} \quad h(w,v)h(v,u)h(w,u) = 1 \tag{36}$$
$$\text{or} \quad h(v,u)h(u,w)h(w,v) = 1 \ , \tag{37}$$

both implying that $\gamma[\alpha[h,e]]_{uvw} \geq 1$, and thus $F \leq 0$. In the latter case,

$$\text{either} \quad h(v,w)h(w,u)h(v,u) = 1 \tag{38}$$
$$\text{or} \quad h(u,v)h(v,w)h(w,u) = 1 \ , \tag{39}$$

both implying again that $\gamma[\alpha[h,e]]_{uvw} \geq 1$ and thus $F \leq 0$.

This concludes the proof of the second part of Theorem 2. ■

3.6 Time Complexity

Running QuickSort does not entail $\Omega(|V|^2)$ accesses to $h_{u,v}$. The following bound on the running time is proven in Section 3.6.

Theorem 6 *The expected number of times QuickSort accesses to the preference function h is at most $O(n \log n)$. Moreover, if only the top k elements are sought then the bound is reduced to $O(k \log k + n)$ by pruning the recursion.*

It is well known that QuickSort on cycle-free tournaments runs in time $O(n \log n)$, where n is the size of the set we wish to sort. That this holds for QuickSort on general tournaments is a simple extension (communicated by Heikki Mannila) which we present it here to keep this presentation self-contained. The second part of the theorem requires some more work.

Proof: Let $T(n)$ be the maximum expected running time of QuickSort on a possibly cyclic tournament on n vertices in terms of number of comparisons. Let $G = (V, A)$ denote a tournament. The main observation is that each vertex $v \in V$ is assigned to the left recursion with probability exactly $\mathrm{outdeg}(v)/n$ and to the right with probability $\mathrm{indeg}(v)/n$, over the choice of the pivot. Therefore, the expected size of both the left and right recursions is exactly $(n-1)/2$. The separation itself costs $n-1$ comparisons. The resulting recursion formula $T(n) \leq n - 1 + 2T((n-1)/2)$ clearly solves to $T(n) = O(n \log n)$.

Assume now that only the k first elements of the output are sought, that is, we are interested in outputting only elements in positions $1, \ldots, k$. The algorithm which we denote by k-QuickSort is clear: recurse with $\min\{k, n_L\}$-QuickSort on the left side and $\max\{0, k - n_L - 1\}$-QuickSort on the right side, where n_L, n_R are the sizes of the left and right recursions respectively and 0-QuickSort takes 0 steps by assumption. To make the analysis simpler, we will assume that whenever $k \geq n/8$, k-QuickSort simply returns the output of the standard QuickSort, which runs in expected time $O(n \log n) = O(n + k \log k)$, within the sought bound. Fix a tournament G on n vertices, and let $t_k(G)$ denote the running time of k-QuickSort on G, where $k < n/8$. Denote the (random) left and right sub-tournaments by G_L and G_R respectively, and let $n_L = |G_L|, n_R = |G_R|$ denote their sizes in terms of number of vertices. Then, clearly,

$$t_k(G) = n - 1 + t_{\min\{k,n_L\}}(G_L) + t_{\max\{0,k-n_L-1\}}(G_R). \tag{40}$$

Assume by structural induction that for all $\{k', n' \colon k' \leq n' < n\}$ and for all tournaments G' on n' vertices,

$$\mathrm{E}[t_{k'}(G')] \leq cn' + c'k' \log k'$$

for some global $c, c' > 0$. Then, by conditioning on G_L, G_R, taking expectations on both sides of (40) and by induction,

$$\mathrm{E}[t_k(G) \mid G_L, G_R] \leq n - 1 + cn_L +$$
$$c' \min\{k, n_L\} \log \min\{k, n_L\} + cn_R \mathbf{1}_{n_L < k-1} +$$
$$c' \max\{k - n_L - 1, 0\} \log \max\{k - n_L - 1, 0\}.$$

By convexity of the function $x \mapsto x \log x$,

$$\min\{k, n_L\} \log \min\{k, n_L\} +$$
$$\max\{k - n_L - 1, 0\} \log \max\{k - n_L - 1, 0\}$$
$$\leq k \log k. \tag{41}$$

Thus,

$$\mathrm{E}[t_k(G) \mid G_L, G_R] \leq n - 1 + cn_L +$$
$$cn_R \mathbf{1}_{n_L < k-1} + c'k \log k. \tag{42}$$

By conditional expectation,

$$\mathrm{E}[t_k(G)] \leq n - 1 + c(n-1)/2 + c'k \log k + c\,\mathrm{E}[n_R \mathbf{1}_{n_L < k-1}].$$

To complete the inductive hypothesis, we need to bound the quantity $\mathrm{E}[n_R \mathbf{1}_{n_L < k-1}]$, which is bounded by $n \Pr[n_L < k-1]$. The event $\{n_L < k-1\}$, equivalent to $\{n_R > n-k\}$, occurs when a vertex of out-degree at least $n - k \geq 7n/8$ is chosen as pivot. For a random pivot $v \in V$, where V is the vertex set of G, $\mathrm{E}[\mathrm{outdeg}(v)^2] \leq n^2/3 + n/2 \leq n^2/2.9$.

Indeed, each pair of edges $(v,u_1) \in A$ and $(v,u_2) \in A$ for $u_1 \neq u_2$ gives rise to a triangle which is counted exactly twice in the cross-terms, hence $n^2/3$ which upper-bounds $2\binom{n}{3}/n$; $n/2$ bounds the diagonal. Thus, $\Pr[\mathrm{outdeg}(v) \geq 7n/8] = \Pr[\mathrm{outdeg}(v)^2 \geq 49n^2/64] \leq 0.46$ (by Markov). Plugging in this value into our last estimate yields

$$\mathrm{E}[t_k(G)] \leq n - 1 + c(n-1)/2 + c'k\log k + 0.46 \times cn,$$

which is at most $cn + c'k\log k$ for $c \geq 30$, as required. ■

4 Lower Bounds

Let r denote the classification regret. Balcan et al. (2007) proved a lower bound of $2r$ for the regret of the algorithm MFAT defined as the solution to the minimum feedback arc-set problem on the tournament V with an edge (u,v) when $h(u,v) = 1$. More precisely, they showed an example of fixed V, h, and bipartite generalized ranking τ^* on V, such that the classification regret of h tends to $1/2$ of the ranking regret of MFAT on V,h. Note that in this case, since τ^* is a fixed function of V, the regret and loss coincide both for classification and for ranking.

Here we give a simple proof of a more general theorem stating that same bound holds for *any* deterministic algorithm, including of course MFAT.

Theorem 7 *For any* deterministic algorithm A *taking as input* $V \subseteq U$ *and a preference function* h *on* V *and outputting a ranking* $\sigma \in S(V)$*, there exists a bipartite distribution* D *on* (V,τ^*) *such that*

$$\mathcal{R}_{rank}(A,D) \geq 2\,\mathcal{R}_{class}(h,D). \tag{43}$$

Note that the theorem implies that, in the bipartite case, no deterministic algorithm converting a preference function into a linear ranking can do better than a randomized algorithm, on expectation. Thus, randomization is essentially necessary in this setting.

The proof is based on an adversarial argument. In our construction, the support of D is reduced to a single pair (V,τ^*) (deterministic input), thus the loss and both the weak and strong regrets coincide and a similar argument applies to the loss function and the weak regret functions.

Proof: Fix $V = \{u,v,w\}$, and let the support of D be reduced to (V,τ^*), where the bipartite generalized ranking τ^* is one that we will select adversarially. Assume a cycle: $h(u,v) = h(v,w) = h(w,u) = 1$. Up to symmetry, there are two options for the output σ of A on V,h.

1. $\sigma(u) < \sigma(v) < \sigma(w)$: in this case, the adversary can choose τ^* corresponding to the partition $V^+ = \{w\}$ and $V^- = \{u,v\}$. Clearly, $\mathcal{R}_{class}(h,D)$ now equals $1/2$ since h is penalized only for misranking the pair (v,w), but $\mathcal{R}_{rank}(A,D) = 1$ since σ is misordering both (u,w) and (v,w).

2. $\sigma(w) < \sigma(v) < \sigma(u)$: in this case, the adversary can choose τ^* corresponding to the partition $V^+ = \{u\}$ and $V^- = \{v,w\}$. Similarly, $\mathcal{R}_{class}(h,D)$ now equals $1/2$ since h is penalized only for misranking the pair (u,w), while $\mathcal{R}_{rank}(A,D) = 1$ since σ is misordering both (u,v) and (u,w).

■

5 Discussion

5.1 History of QuickSort

The textbook algorithm, by now standard, was originally discovered by Hoare (1961). Montague and Aslam (Montague & Aslam, 2002) experimented with QuickSort for information retrieval (IR) by aggregating rankings from different sources of retrieval. They claimed an $O(n\log n)$ time bound on the number of comparisons, although the proof seemed to rely on the folklore QuickSort proof without addressing the non-transitivity problem. They proved certain combinatorial bounds on the output of QuickSort and provided an empirical justification of its IR merits. Ailon et al. (2005) also considered the rank aggregation problem and proved theoretical cost bounds for many ranking problems on weighted tournaments. They strengthened these bounds by considering non-deterministic pivoting rules arising from solutions to certain ranking LP's. This work was later extended by Ailon (2007) to deal with rankings with ties, in particular, top-k rankings. Hedge et al.(2007) and Williamson and van Zuylen (2007) derandomized the random pivot selection step in QuickSort for many of the combinatorial optimization problems studied by Ailon et al.

5.2 The decomposition technique

The technique developed in Lemma 3 is very general and can be used for a wide variety of loss functions and variants of QuickSort involving non-deterministic ordering rules (Ailon et al. 2005; 2007). Such results would typically amount to bounding $\beta[X]_{uvw}/\gamma[Z]_{uvw}$ for some carefully chosen functions X, Z depending on the application.

5.3 Combinatorial Optimization vs. Learning of Ranking

QuickSort, sometimes referred to as FAS-Pivot in that context, was used by Ailon et al. (2005; 2007) to approximate certain NP-Hard weighted instances of the problem of minimum feedback arcset in tournaments (Alon, 2006). There is much similarity between the techniques used in that work and those of the analyses of this work, but there is also a significant difference that should be noted.

In the minimum feedback arc-set problem, we are given a tournament G and wish to find an acyclic tournament H on the same vertex set minimizing $\Delta(G,H)$, where Δ counts the number of edges pointing in opposite directions between G,H (or a weighted version thereof). However, the cost we are considering is $\Delta(G,H_\sigma)$ for some fixed acyclic tournament H_σ induced by some permutation σ (the ground truth). In this work, we showed in fact that if G' is obtained from G using QuickSort, then $\mathrm{E}[\Delta(G',H_\sigma)] \leq 2\Delta(G,H_\sigma)$ for *any* σ (Theorem 1). If H is the optimal solution to the (weighted) minimum feedback arc-set problem corresponding to G, then it is easy to see that $\Delta(H,H_\sigma) \leq \Delta(G,H) + \Delta(G,H_\sigma) \leq 2\Delta(G,H_\sigma)$. However, recovering G is NP-Hard in general. Approximating $\Delta(G,H)$ modulo a constant factor $1+\varepsilon$ using an acyclic tournament H', as in the combinatorial optimization world, only guarantees a constant factor of $2+\varepsilon$:

$$\Delta(H',H_\sigma) \leq \Delta(G,H') + \Delta(G,H_\sigma) \leq (1+\varepsilon)\Delta(G,H) + \Delta(G,H_\sigma) \leq (2+\varepsilon)\Delta(G,H_\sigma)\ .$$

Thus, this work also adds a significant contribution to (Ailon et al., 2005; Ailon, 2007; Kenyon-Mathieu & Schudy, 2007).

5.4 Weak vs. Strong Regret Functions

For the proof of the regret bound of Theorem 2 we used the fact that the minimizer $\tilde{h}$ in the definition (5-6) of $\mathcal{R}'_{class}$ could be determined independently for each pair $u, v \in U$, using (9). This could also be done for strong regrets if the distribution D on V, τ^* satisfied the following pairwise IIA condition.

Definition 8 *A distribution D on subsets $V \subseteq U$ and generalized rankings τ^* on V satisfies the* pairwise independence on irrelevant alternatives *(pairwise IIA) if for all $u, v \in U$ and for any two subsets $V_1, V_2 \supseteq \{u, v\}$,*

$$E_{\tau^*|V_1}[\tau^*(u,v)] = E_{\tau^*|V_2}[\tau^*(u,v)] \ .$$

Note: We chose the terminology IIA to match that used in Arrow's seminal work (Arrow, 1950) to describe a similar notion.

When pairwise IIA holds, the average ground truth relation between u and v, conditioned on u, v included in V, is independent of V.

Recall that a bipartite τ^* is derived from a pair σ^*, ω, where ω is defined using a term $1/m^- m^+$, for compatibility with the definition of AUC. The numbers m^+ and m^- depend on the underlying size of the positive and negative sets partitioning of V and therefore cannot be inferred from (u, v) alone. Thus, in the standard bipartite case, the pairwise IIA assumption is not natural. If, however, we replaced our definitions in the bipartite case and used the following:

$$\omega(i,j) = \begin{cases} 1 & (i \leq m^+) \wedge (j > m^+) \\ 1 & (j \leq m^+) \wedge (i > m^+) \\ 0 & \text{otherwise,} \end{cases} \qquad (44)$$

instead of (2), then it would be reasonable to believe that pairwise IIA does hold in the bipartite case. In fact, it would be reasonable to make the stronger assumption that for any fixed $u, v \in U$ and $V_1, V_2 \supseteq \{u, v\}$ the distribution of the random variables $\tau^*(u,v)|V_1$ and $\tau^*(u,v)|V_2$ are equal. This corresponds to the intuition that when comparing a pair u, v in a context of a set V containing them, human labelers are not as influenced by the irrelevant information $V \backslash \{u, v\}$ as they would be by $V \backslash \{u\}$ if asked to evaluate single elements u. The irrelevant information in V is often referred to as *anchor* in experimental psychology and economics (Ariely et al., 2008).

Our regret bounds would still hold if we used (44), but we chose (2) to present our results in terms of the familiar average pairwise misranking error or AUC loss.

Another possible assumption allowing usage of strong regrets is to let the preference function learned in the first stage depend on V. This is the assumption implicitly made by Balcan et al. (2007) (based on our private communication). We do not further elaborate on this assumption.

6 Conclusion

We described a reduction of the learning problem of ranking to classification. The efficiency of this reduction makes it practical for large-scale information extraction and search engine applications. A finer analysis of QuickSort is likely to further improve our reduction bound by providing a concentration inequality for the algorithm's deviation from its expected behavior using the confidence scores output by the classifier. Our reduction leads to a competitive ranking algorithm that can be viewed as an alternative to the algorithms previously designed for the score-based setting.

7 Acknowledgments

We thank Alina Beygelzimer and John Langford for helpful discussions. Mehryar Mohri's work was partially funded by the New York State Office of Science Technology and Academic Research (NYSTAR).

References

Agarwal, S., Graepel, T., Herbrich, R., Har-Peled, S., & Roth, D. (2005). Generalization bounds for the area under the roc curve. *Journal of Machine Learning Research, 6*, 393–425.

Agarwal, S., & Niyogi, P. (2005). Stability and generalization of bipartite ranking algorithms. *COLT* (pp. 32–47).

Ailon, N. (2007). Aggregation of partial rankings, p-ratings and top-m lists. *SODA*.

Ailon, N., Charikar, M., & Newman, A. (2005). Aggregating inconsistent information: ranking and clustering. *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005* (pp. 684–693). ACM.

Alon, N. (2006). Ranking tournaments. *SIAM J. Discrete Math., 20*, 137–142.

Ariely, D., Loewenstein, G., & Prelec, D. (2008). Coherent arbitrariness: Stable demand curves without stable preferences. *The Quarterly Journal of Economics, 118*, 73–105.

Arrow, K. J. (1950). A difficulty in the concept of social welfare. *Journal of Political Economy, 58*, 328–346.

Balcan, M.-F., Bansal, N., Beygelzimer, A., Coppersmith, D., Langford, J., & Sorkin, G. B. (2007). Robust reductions from ranking to classification. *COLT* (pp. 604–619). Springer.

Clémençon, S., & Vayatis, N. (2007). Ranking the best instances. *Journal of Machine Learning Research, 8*, 2671–2699.

Cohen, W. W., Schapire, R. E., & Singer, Y. (1999). Learning to order things. *J. Artif. Intell. Res. (JAIR), 10*, 243–270.

Coppersmith, D., Fleischer, L., & Rudra, A. (2006). Ordering by weighted number of wins gives a good ranking for weighted tournamnets. *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*.

Cortes, C., Mohri, M., & Rastogi, A. (2007a). An Alternative Ranking Problem for Search Engines. *Proceedings of the 6th Workshop on Experimental Algorithms (WEA 2007)* (pp. 1–21). Rome, Italy: Springer-Verlag, Heidelberg, Germany.

Cortes, C., Mohri, M., & Rastogi, A. (2007b). Magnitude-Preserving Ranking Algorithms. *Proceedings of the Twenty-fourth International Conference on Machine Learning (ICML 2007)*. Oregon State University, Corvallis, OR.

Cossock, D., & Zhang, T. (2006). Subset ranking using regression. *COLT* (pp. 605–619).

Crammer, K., & Singer, Y. (2001). Pranking with ranking. *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]* (pp. 641–647). MIT Press.

Freund, Y., Iyer, R. D., Schapire, R. E., & Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, *4*, 933–969.

Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*.

Hedge, R., Jain, K., Williamson, D. P., & van Zuylen, A. (2007). "deterministic pivoting algorithms for constrained ranking and clustering problems". *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*.

Hoare, C. (1961). Quicksort: Algorithm 64. *Comm. ACM*, *4*, 321–322.

Joachims, T. (2002). Optimizing search engines using clickthrough data. *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 133–142). New York, NY, USA: ACM Press.

Kenyon-Mathieu, C., & Schudy, W. (2007). How to rank with few errors. *STOC '07: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing* (pp. 95–103). New York, NY, USA: ACM Press.

Lehmann, E. L. (1975). *Nonparametrics: Statistical methods based on ranks*. San Francisco, California: Holden-Day.

Montague, M. H., & Aslam, J. A. (2002). Condorcet fusion for improved retrieval. *Proceedings of the 2002 ACM CIKM International Conference on Information and Knowledge Management, McLean, VA, USA, November 4-9, 2002* (pp. 538–548). ACM.

Rudin, C., Cortes, C., Mohri, M., & Schapire, R. E. (2005). Margin-based ranking meets boosting in the middle. *Learning Theory, 18th Annual Conference on Learning Theory, COLT 2005, Bertinoro, Italy, June 27-30, 2005, Proceedings* (pp. 63–78). Springer.

Williamson, D. P., & van Zuylen, A. (2007). "deterministic algorithms for rank aggregation and other ranking and clustering problems". *Proceedings of the 5th Workshop on Approximation and Online Algorithms (WAOA) (to appear)*.